

PROGRAMMAZIONE 2

canale A-D 2007-2008

Prof. Francesco PARISI PRESICCE

STUDIO Dipartimento di Informatica

Via Salaria 113 3° piano

TELEFONO 06 4991 8514

EMAIL parisi AT di PUNTO uniroma1 PUNTO it

RICEVIMENTO Martedì e Venerdì 14:30-16:30

ESERCITAZIONI dott. Lorenzo Carlucci

TESTO CONSIGLIATO:

Al Kelley, I. Pohl [C, didattica e programmazione](#), Addison-Wesley,
traduzione italiana a cura del Prof. G. Pighizzini, Univ. Milano

Programma (preliminare) 2007-08

- **Ricorsione.** Richiami di programmazione ricorsiva: fondamenti della ricorsione; lo stack delle chiamate ricorsive; esempi di programmi ricorsivi. La "tail recursion". Strutture dinamiche ricorsive: le liste.
- **Cenni di complessità di spazio e di tempo per gli algoritmi e i programmi.** Valutazione asintotica e miglioramento delle costanti moltiplicative. Confronto tra soluzioni iterative e ricorsive.
- **Tecniche per la scrittura di programmi corretti.** L'importanza della specifica precisa di un programma. Precondizioni e postcondizioni delle funzioni. Uso di assert per la loro verifica. Strutturazione dei programmi: sottoprogrammi e moduli. Tipi di dato astratti. L'information hiding. Pile e code: definizione e implementazione mediante liste e vettori. Invarianti come strumenti di progettazione di cicli. Applicazione del metodo al problema dell'ordinamento.
- **Alberi binari.** definizione e implementazione. Proprietà degli alberi binari: altezza, foglie, relazione tra numero dei nodi e altezza in un albero binario qualunque, alberi pieni, completi, criteri di bilanciamento. Alberi binari di ricerca.
- **Tecniche di testing.** La scelta dei dati di test negli approcci Black-box e glass-box.

Nel passato (on tanto remoto!) sono stati prodotti moltissimi programmi che

- terminano imprevedibilmente (crash),
- non producono risposte corrette (pieni di bugs),
- sono costosi da correggere o modificare, e
- contengono errori nascosti (timebombs) che si manifesteranno prima o poi

Requisiti dei programmi

correttezza	proprietà oggettiva
efficienza	proprietà oggettiva
facilità di lettura (quindi comprensione)	non quantificabile
facilità di manutenzione	“
robustezza	“
portabilità	“
riusabilità	“

Scrivere programmi è MOLTO facile

La cosa difficile (!) è scrivere programmi **corretti**

Cosa fare:

Progettare accuratamente il programma scegliendo le strutture dati più convenienti, suddividendolo in moduli coerenti per obiettivo (per esempio tutte le funzioni di I/O,...)

Per ogni compito che deve essere eseguito scrivere la relativa funzione in pseudocodice e verificarne la correttezza

Cosa non fare:

Scrivere il programma senza preoccuparsi dello stile o della strutturazione e affidare ogni verifica della correttezza al testing